Encrypting Blaise Data on Network Servers

John Mamer, Leonard Hart and Josh Rozen, Mathematica Policy Research, Inc.

1. Background

Identity theft and privacy issues have brought increased scrutiny of the technology and procedures by which confidential data about individuals are protected. High-profile cases over the last several years, in which confidential data have been stolen or inadvertently released, have dramatically increased both awareness of these issues and pressure to take all conceivable precautions to prevent such incidents. These cases and the increased scrutiny they have brought to security and confidentiality issues have affected virtually every public and private institution in the United States.

One of these cases—the theft of a laptop containing personal records of more than 26.5 million veterans from the U.S. Department of Veterans Affairs (VA)—brought increased focus by agencies within the federal government on the issue of data "at rest." The level of protection implied by this concern goes beyond the two issues that have plagued the information technology industry since the inception of the internet: (1) security of data transmitted across the internet ("in motion"), and, (2) unauthorized intrusion into local area networks maintaining secure data ("hacking"). Security directives from a number of government agencies—backed by National Institute of Standards and Technology (NIST) guidelines—are increasingly calling for the encryption of all data stored on electronic media.

Initially, federal government agencies focused their attention on the vulnerability of laptop data exposed by incidents such as the VA theft cited above. However, they are now expanding their focus to network servers and other network-connected devices behind secure network firewalls. Mathematica Policy Research, Inc. (MPR) collects data—often very sensitive data—on behalf of the federal government and security has always been a major concern. New security directives from the federal government impact the systems we develop and/or use to collect such data. Since Blaise is one of those systems, we have investigated the means by which Blaise data can be held in encrypted form. This paper presents the results of our investigation.

2. Full Disk vs. Content Encryption

Our investigation into encrypting Blaise data began with identifying the differences between full disk encryption and content encryption. An understanding of these differences is important to devising an approach to Blaise encryption. In this section, we summarize our findings.

2.1 Full Disk Encryption

In full disk encryption, the entire contents of a data storage device are kept in an encrypted format. Full disk encryption is most easily understood in the context of the theft of the VA laptop noted above. Even with strong passwords in use at the operating system level, data on an unencrypted drive might easily be accessed by simply mounting that drive as a non-system drive on a different computer.

A fully encrypted hard drive renders the laptop's drive useless to any user who does not have the username and password (or encryption key). Authentication components of the software are installed in the system drive's Master Boot Record (MBR); thus there is no way of circumventing the authentication process when the laptop is booted. When a valid username and password are supplied, the drive contents (starting with operating system files) are decrypted into memory as needed. Data are written (encrypted) and read (decrypted) to and from the drive via what amounts to system level i/o drivers, transparent to users and programs. Other than a small performance penalty, the machine runs as it normally would.

It is now standard practice at MPR to run full disk encryption (256-bit AES) on all corporate laptops. As a priority, we encrypted all laptops used in our Computer Assisted Personal Interviewing (CAPI) applications. The CAPI laptops run the Blaise software along with SurveyTrak software developed by the Survey Research Center at the University of Michigan. Identifying information on the respondents to be surveyed, as well as information gathered from these respondents using the Blaise system, is stored in a Sybase database in encrypted format. This provides reasonable protection to data on the laptops. However, the Sybase encryption—at least the version of Sybase in the SurveyTrak implementation used by MPR—does not conform to the Federal Information Processing Standards (FIPS) 140-2 certification requirements from the NIST. Adherence to FIPS 140-2 (Security Requirements for Cryptographic Modules) is now being required by many of our clients.

MPR evaluated two products for implementing full disk encryption on its CAPI laptops—SafeBoot (from SafeBoot International) and Pointsec (from Pointsec Mobile Technologies, Inc.). Both worked as advertised and would have met all government standards with one exception—Pointsec was not FIPS 140-2 certified for the version of Windows that was being run on our older CAPI laptops. Primarily for this reason, we eventually chose SafeBoot.

Neither Pointsec nor SafeBoot was designed with network server drives in mind, but follow-up discussions with sales people from the respective organizations indicated that the products would work in the network environment. However, for the purposes of these applications, the Network Operating System (NOS) is the "user" of the network drives. Once the NOS supplies the password, just as with laptop drives, the encryption of data becomes transparent to the "user." While this is certainly good from the standpoint of ease of implementation, in the network environment the only event that full disk encryption protects against is the theft of a network server drive. This limitation led to the investigation of content encryption, as described in the next section.

2.2 Content Encryption

The term "content encryption" appears to be trademarked by SafeBoot but is often used in conjunction with the products supplied by a number of other vendors, such as PGP, McAfee, and Coreguard. These products vary in their details but share a few common features:

- Data can be encrypted on network drives at the file or folder level

- Authentication through an “encryption engine” or an “encryption server” is required before data can be passed to a network user

- Client-side software is used to decrypt the data when it is accessed on the user’s machine, again requiring authentication when decryption is performed

The disadvantage of content encryption is the significantly increased cost of purchase, implementation, and administration compared to full disk encryption. The advantage is that content encryption provides protection against much more than the theft of a network server drive, including the following types of protection:

- Data are encrypted as they are transmitted across the local area networks

- Data can be protected even from network administrators

- As one vendor (PGP) puts it, “the encryption follows the data.” For example, data can be downloaded from a network drive to a USB drive by an authenticated user but it will be in encrypted form. It can subsequently only be accessed by a user who is authenticated through the encryption server.

3. Blaise and Network Server Encryption Technologies

With the previous section as background, this section describes our investigation into the specific encryption technologies that we considered for use in storing and processing encrypted Blaise files on network servers.

3.1 Windows Encrypted File System

The Windows Encrypted File System (EFS) from Microsoft is best classed as a content encryption approach to encryption. As with many “free” Microsoft products, it does not provide much of the functionality that third party vendors provide but it has some of the attributes described above for content encryption. EFS can be used on a desktop with the Windows operating system (XP or higher is best) or on Windows-based servers to encrypt individual folders or files. After encryption, only authorized users may access or decrypt the files.

Our initial expectation was that EFS would provide the best approach to encryption for Blaise files. This was based on our success using EFS for encrypting SQL Server databases, which our federal government clients are also requiring. There are three salient aspects of utilizing EFS for encrypting SQL Server databases:

- An SQL Server database is a single file in the Windows operating system and EFS is easily applied to that single file.

- The SQL Server database engine is the “user” of that file. Once SQL Server has performed the decryption, all programs run transparently against the database (that is, no code changes are required to switch from running an unencrypted database to running an encrypted database).

- Performance penalties are documented in the 5 to 10 percent range and we observed no appreciable degradation in response times for online, interactive applications using an encrypted SQL Server database.

For encrypting Blaise applications, it quickly became apparent that EFS was *not* a workable solution. Two problems presented themselves immediately:

- With EFS, users can encrypt data at the file or folder level, but they can only grant access at the file level. Whereas with SQL Server we only had to encrypt a single file, with Blaise many files are involved: the .bdb file, audit-trails, batch and overnight files, lock files, temp files, and so on. We determined that it was not feasible to administer individual file rights under EFS to multiple Blaise files when we have many Blaise applications being implemented in a constantly revolving environment.

- With SQL Server—a true client server application—only one "user" needs to be authenticated in order to provide decrypted data to multiple application users. That user is the SQL Server application itself. With Blaise, each individual Blaise user (interviewer) requires authentication, again adding to administrative burden.

3.2 Novell Network Operating System Version 6.5

MPR has utilized Novell as the platform for its Blaise applications. Novell does not have the security vulnerabilities associated with Microsoft, largely because it is not as ready a target for hackers. Also, Blaise utilizes a network server as a file server and Novell is a very efficient file server. Market pressures and other factors will, in the long run, force MPR to move to the Windows server platform but this will be a difficult migration. Even if we had found Windows Encrypted File System to be a good solution to the encryption problem with Blaise, the difficulty of migration and the need for a quick solution to the encryption requirements would have presented a major challenge.

Version 6.5 of the Novell Network Operation System (NOS) provides for the encryption of network drives. Our initial tests show that performance of Blaise on encrypted network drives utilizing Version 6.5 of the Novell NOS is acceptable. Our plan is to migrate any Blaise application requiring encryption on network servers to a server running Novell NOS Version 6.5. These applications and surveys for the most part are currently running in Blaise Version 4.7. In this particular approach, Blaise Version 4.8 would appear to provide no particular advantage, so we are not planning to utilize Version 4.8 as a part of the solution to Blaise encryption.

3.3 Content Encryption

In the long term (but it is not too far away), we expect that the federal government will require content encryption for all confidential data including any data in Blaise files or related processes. All such files will need to be encrypted using some form of content encryption. This will secure data that are transmitted across the local area network, it will protect data from network administrators, and it will protect data that are downloaded onto removable media. Unlike Windows EFS, which only allows users to grant and revoke rights at the file level, commercial content encryption packages allow these facilities at the folder level. With these facilities, the administrative burden will be significant but it should be manageable.

3.4 Encryption and Decryption Using the Blaise Software

The feasibility of storing Blaise files as encrypted files, and handling encryption and decryption within the Blaise software, was raised with the Blaise team at the Blaise Corporate User Group meeting held in Dusseldorf earlier this year. The team's response was that this was feasible from a development standpoint but that it could seriously impact usability and ease of implementation of Blaise applications, because of the login and password requirements that would be required for both interviewers and application programs (written in Manipula or any other application) that access Blaise data. Based on this discussion, it was decided to drop this requirement from the "wish list" for future Blaise releases. Given the clearer definition of requirements that we feel we have gained over that last six months and the administrative costs of the content encryption approach that is likely to be required in the future, we feel that it would be worth reconsidering this option, but, given that content encryption may be "in our future" for many other reasons, encryption through the Blaise software may still be deemed a less attractive option.

4. Conclusions and Next Steps

As noted above, MPR believes that content encryption of all confidential data, including the data used to field a Blaise survey and the data collected by a Blaise instrument, will be required in most or all federal government applications/surveys. MPR will continue its investigation of third party vendors providing content encryption solutions. Once a vendor is selected, the implementation of encryption in Blaise will be a difficult conversion where not only the .bdb file but all files related to Blaise processing containing confidential data will have to be reviewed. By segregating the .bdb file to a separate server, Blaise Version 4.8 may help in the solution of the encryption problem and this will be investigated. We would welcome a review by the Blaise team of the requirements and issues presented in this paper in order to reconsider whether an encryption/decryption facility within the Blaise components presents a better option. However, in sum, the impact of new security requirements on the Blaise application and any other applications that store and process confidential data should not be underestimated.